

Applicant	:	Mark E. Madsen, et al.
Appl. No.	:	10/817,517
Examiner	:	not yet assigned
Docket No.	:	13508.4001

Amendments to the Specification

Please replace paragraphs [0108], [0109], and [0133] with the following amended paragraphs:

[0108] Turning to FIG. 12, acquisition of an abstract syntax tree (Phase I) begins at step 1210 with identifying a set of “command roots” – e.g. the first several tokens that make up a command in a line-oriented configuration language. Roots serve to define the starting point and “breadth” of a syntax tree search. As an example, we will use the command “ip route 10.0.0.0 255.0.0.0 192.168.1.100 10”, which adds a static route to the IP network 10.0.0.0 via the network link located at 192.168.1.100, with a “distance” (or preference, essentially) of 10. The “command root” we will start with is the partial command “ip route,” and our goal will be to automatically discover the syntax of “ip route” commands in IOS 12.2. Starting points are arbitrary – we could equally begin, for example, with the command root “ip” and discover all commands that follow this root. When Grammar Builder 45 is activated within the system, the “command root” will be provided by the candidate component 58 being resolved into a resolved grammar 70.

[0109] In order to discover the allowable syntax, at step ~~1210~~1220 an embodiment of the invention uses a vendor-specific algorithm to “walk” any command completion or command-line help available, and view the options available at each point in the command structure. This algorithm is generically called “WalkerViewer.” In an embodiment disclosed below, the WalkerViewer algorithm for Cisco’s IOS operating system is discussed in detail. Other operating systems and other vendors are also supported by modified version of the WalkerViewer algorithm. For example, the WalkerViewer algorithm for Cisco’s Catalyst operating system differs only in minor details.

[0133] The result of processing of the syntax tree 80 is shown in Table 65 below. In contrast to the syntax tree 80, note that there is only one “path” through the grammar rule, as defined by end-of-line characters. Multiple possible paths through the tree of SyntaxNodes have been collapsed into alternate or optional sections. The top-level rule is marked as “FINAL,” denoting a grammar section which corresponds to a single user-perceptible command – in this case, our “ip route...” example.